Sun Enterprise Server Alternate Pathing 2.3 Reference Manual

Adobe PostScript™

Please Recycle

Contents

Maintenance Commands

NAME	Intro – AP administration
DESCRIPTION	This section describes commands, scripts, and programs executed in the alternate pathing environment.
LIST OF COMMANDS	

ap**(1M)**   alternate pathing

ap_daemon**(1M)**   alternate pathing daemon

apboot**(1M)**   set up system files for boot metadisk

apcheck**(1M)**   determine accessibility of a metadisk

apconfig**(1M)**   display and manage AP configuration

apdb**(1M)**   manage AP database

apdisk**(1M)**   manage disk pathgroups

apinst**(1M)**   identify disk host adapter instances, /dev/dsk targets

apnet**(1M)**   manage network pathgroups

NAME | ap – alternate pathing

DESCRIPTION | Alternate pathing (AP) enables you to define and control alternate physical paths to peripheral devices. If a path to a device becomes unavailable, your Sun server can use an alternate path.

SEE ALSO | *Sun Enterprise Server Alternate Pathing 2.3 User Guide*

`apconfig`(1M), `apdb`(1M), `apdisk`(1M), `apnet`(1M)

NAME | apboot – set up system files for boot metadisk

SYNOPSIS | **apboot** [−n] [−o] [−k *system-name*] [−v *vfstab-name*] *device*

apboot [−n] [−o] [−v *vfstab-name*] [−m *metadisk*]

apboot [−n] [−o] [−v *vfstab-name*] [−u *metadisk*]

DESCRIPTION | Use /usr/sbin/apboot to edit /etc/vfstab and /etc/system to make the system bootable from either the boot disk file systems on an AP metadisk or the boot disk file systems on a physical disk device that is not alternately pathed. The apboot(1M) command enables AP to manage a mirrored boot device when both that boot device and its mirror are under AP control.

In addition to editing /etc/vfstab/ and /etc/system,apboot(1M) checks the current configuration of system swap and dump devices. If either is configured as a partition of the boot disk, apboot(1M) calls swap(1M) or dump(1M), as appropriate, to ensure that swap and dump devices are consistent with the boot device.

Finally, apboot(1M) modifies the boot-device property of the OpenBoot PROM (OBP) so that all paths to the boot-device (and its mirror, if applicable) are listed in the order in which they will be tried during an automatic or manual reboot. The default order in an alternately pathed mirrored system is: Primary root, Primary mirror, Alternate root, Alternate mirror. This is a change from previous versions of AP to improve redundancy and serviceability. The default order in an alternately pathed, non-mirrored system is: Primary root, Alternate root. Use the eeprom(1M) command to view the boot-device property setting.

OPTIONS | The following options are supported:

−n

  Print what would be done without actually doing it.

−k *system-name*

  Edit *system-name* instead of the default /etc/system file.

−v *vfstab-name*

  Use or edit *vfstab-name* instead of the default /etc/vfstab table of file system defaults.

−m *metadisk*

Enable boot mirror support for the specified AP metadisk.

−o

Suppress automatic update of the `boot-device` property in OpenBoot
PROM.

−u *metadisk*

Disable boot mirror support for the specified AP metadisk.

EXAMPLES

EXAMPLE 1   Using `apboot` with Metadisks

The following command edits `/etc/system` and `/etc/vfstab` to specify
that the boot-disk file systems are now on metadisk `mc3t0d0`.

   apboot mc3t0d0

EXAMPLE 2   Using `apboot` with Physical Devices

The following command edits `/etc/system` and `/etc/vfstab` to specify
that the boot-disk file systems are now under the physical path
`/dev/dsk/c3t0d0`.

   apboot c3t0d0

EXAMPLE 3   Using `apboot` with Mirrored Devices

The following commands edit `/etc/system` and `/etc/vfstab` to specify
that the boot disk file systems are now on metadisk `mc3t0d0`, with a mirror
on `mc1t0d1`.

   apboot mc3t0d0
   apboot −m mc1t0d1

EXAMPLE 4   Using `apboot` to Disable Mirrored Devices

The following command disables AP support for the mirror device created in
the previous example.

   apboot −u mc1t0d1

FILES

The following files are used by this utility:

`/etc/system`   Kernel patch file

`/etc/vfstab`   Table of file system defaults

SEE ALSO   boot(1M), dumpadm(1M), eeprom(1M), swap(1M), system(4), and
vfstab(4) in the *SunOS Reference Manual*

NAME | apcheck – determine accessibility of a metadisk

SYNOPSIS | **apcheck** *special*

DESCRIPTION | `/sbin/apcheck` ascertains whether a metadisk is usable. If it is able to locate dual paths, `apcheck` exits with a zero status; if not, it exits with a non-zero status.

CAUTION | Do not execute `apcheck` on the command line; it is intended for use only by other commands or by authorized service providers.

OPTIONS | The following options are supported:
special | This option represents the device node to be checked. This device node may reside under `/dev/ap/dsk` or `/dev/ap/rdsk`.

NAME | apconfig – display and manage AP configuration

SYNOPSIS | **apconfig** −D

apconfig −F

apconfig −N [−u]

apconfig −P *metanetwork_interface* −a *physical_interface*

apconfig −P *primary_disk_controller* −a *physical_disk_controller*

apconfig −R

apconfig −S [−u]

DESCRIPTION | The /usr/sbin/apconfig command displays and helps you manage the
alternate pathing (AP) 2.3 system configuration.

OPTIONS | The following options are supported:

−D

Display location and status information for all known copies of the host
database.

−F

Force the state (attached or detached) of every committed path group
alternate to match the physical state of the system. Use this option if the two
states differ. It refreshes the dynamic reconfiguration (DR) flags for every
disk I/O port and physical network interface defined for all committed path
groups.

−N **[−u]**

Display network AP information only. For each path group, apconfig(1M)
−N displays the metanetwork interface and the corresponding physical
network interfaces.

If you specify the −u option, apconfig(1M) displays uncommitted path
group information only. If you do not specify the −u option, apconfig(1M)
displays committed path group information only. See "Character flags after
metadevice names" and "Character flags after physical device paths", below.

−P *metanetwork_interface* −a *physical_interface*

Switch to the new physical path specified by −a for the metanetwork specified by −P.

−P *primary_disk_controller*− a *physical_disk_controller*

Switch to the new physical path specified by −a for the metadisk(s) associated with the primary path specified by −P.

Note - Pathnames for disk controllers consist of the I/O port name and the instance number separated by a colon (:) , for example, pln:0. This naming convention differs from previous versions of AP. The addition of the colon (:) delimiter allows AP to provide support for a wider variety of disk controllers. This change does not apply to network pathnames.

−R

Rebuild the metadisk device nodes in /dev/ap/dsk and /dev/ap/rdsk. The apconfig command creates links to /devices for all committed disk path groups in the database.

Note - You must execute drvconfig -i ap_dmd before you can execute apconfig -R. See drvconfig(1M) and ap_dmd(7D).

−S **[**−u**]**

Display alternate pathing information for disk path groups only. For each path group, apconfig(1M) shows the names for the metadisk, its physical devices, and the disk I/O ports through which each physical device is accessed.

If you specify the −u option, apconfig(1M) displays only uncommitted path group information. Otherwise, it displays only committed path group information. See "Character flags after metadevice names" and "Character flags after physical device paths", below.

Character flags after metadevice names

When you specify −N or −S, one or more of the following letters may be displayed after each metanetwork or metadisk name:

D   Marked for deletion. The metadisk or metanetwork remains in the database and continues to be used by AP until a commit is done. See apdb(1M).

U	Uncommitted. Note that you cannot use a metadisk or metanetwork until a commit has been done.
R	Marked for use as a root device (-S only).
M	Marked as the mirror for a boot device (-S only).
X	The physical paths for this metadisk lead to different disks–that is, different supported dual-ported disk array storage devices (-S only). For a list of supported devices, see the *Sun Enterprise Server Alternate Pathing 2.3 User Guide*

Character flags after physical device paths.

When you specify -N or -S, one or more of the following letters may be displayed after each physical network path or physical disk I/O port path:

N	Automatic switching is not allowed for this physical device.
X	The physical paths for this metadisk lead to different disks (that is, different storage array device disks). (-S only.)
A	The active alternate (to select another alternate, use the −P and −a options)
DR	Marked as being drained by the DR daemon. A switch cannot be made to a device path in this state. See the *Sun Enterprise Server Alternate Pathing 2.3 User Guide*.
DE	Marked as detached by the DR daemon.
P	The primary path for disk path groups. This primary path cannot be changed. (-S only.)
T	Path has been tried as active.
O	Marked as offline. See apdisk(1M) and apnet(1M).

EXAMPLES

EXAMPLE 1   Displaying Committed Disk Path Groups

The following command displays all of the committed disk path groups in the AP database.

```
# apconfig −S
c6      pln:0  P
c2      pln:3  A
        metadiskname(s):
                mc2t5d0
                mc2t4d0  R
```

```
                              mc2t3d0
                              mc2t2d0
                              mc2t1d0
                              mc2t0d0
```

EXAMPLE 2   Displaying Uncommitted Network Path Groups

The following example displays all of the uncommitted network path groups
in the AP database:

```
 # apconfig −N −u
 metanetwork:   mether0  U
 physical devices:
         qfe1  A
         qfe0
```

EXAMPLE 3   Switching the Active Path Group

The following example switches the active alternate of the disk path group for
which the primary path is pln:1. The new active alternate of that path group
is pln:0.

```
 # apconfig −P pln:1 −a pln:0
```

EXAMPLE 4   Switching the Network Path Group

The following example switches the active alternate of the network path group
identified by the metanetwork interface mether0. The new active alternate of
that network path group is qe1.

```
 # apconfig −P  mether0 −a  qe1
```

EXAMPLE 5   Displaying AP Database Information and Location

The following example displays the location and status information of all
known copies of the AP database.

```
 # apconfig −D
 path: /dev/rdsk/c3t3d0s1
 major: 32
 minor: 145
 timestamp: Wed Sep 28 18:45:58 1994
 checksum: 2636010350
 default: yes
 corrupt: no
 inaccessible: no

 path: /dev/rdsk/c3t3d0s6
 major: 32
 minor: 150
 timestamp: Wed Sep 28 18:50:43 1994
 checksum: 2636010350
```

```
default: no
synced: yes
corrupt: no
inaccessible: no
```

SEE ALSO   *Sun Enterprise Server Alternate Pathing 2.3 User Guide*

apdb(1M), apdisk(1M), apnet(1M), ap_dmd(7D), mether(7D), mfddi(7D) in this reference manual

drvconfig(1M)in the *SunOS Reference Manual*

NAME	ap_daemon – alternate pathing daemon
SYNOPSIS	**ap_daemon**
DESCRIPTION	The `/usr/sbin/ap_daemon` is an RPC program that provides the interface to the alternate pathing (AP) driver.
Configuration Information	The `ap_daemon` RPC program name is AP_SVR; its RPC program number is 300473; and, its underlying protocol is TCP. It is invoked as an `inetd` server by using the TCP transport. Access to the daemon is available only for requests originating from `localhost` with UID zero.

The entry for the daemon in the `/etc/inetd.conf` file is:

```
300473/1 tli rpc/tcp wait root /usr/sbin/ap_daemon ap_daemon
```

SEE ALSO	*Sun Enterprise Server Alternate Pathing 2.3 User Guide*

`apconfig`(1M), `apdb`(1M), `apdisk`(1M), `apnet`(1M)

NAME	apdb – manage AP database
SYNOPSIS	**apdb** −c *raw_disk_slice* [−k *system_file*] [−f]
	apdb −d *raw_disk_slice* [−k *system_file*] [−f]
	apdb −m *major_number* −n *minor_number* [−f]
	apdb −C
	apdb −Z
DESCRIPTION	The /usr/sbin/apdb command helps you manage the AP database.
OPTIONS	The following options are supported:

−c *raw_disk_slice*

Create a database copy on the specified raw disk slice. You can create up to ten copies of the database. The minimum slice size is 300-KBytes.

−d *raw_disk_slice*

Delete a database copy from the specified raw disk slice.

−f

Force the creation or deletion of the specified database. This option is required for creating the first copy of the database and for deleting each of the last two copies of the database. If you try to delete a database copy without this option when fewer than two database copies exist, AP displays an error message.

−k *system_file*

Patch the database copy information to the kernel file *system_file*, rather than the default file, /etc/system.

−m *major_number* −n *minor_number*

Remove a database copy by specifying its location as a major-minor pair. Use −m to specify the major number and −n for the minor number. This option pair is useful when there is no path to the database because the device no longer exists.

−C

Commit all uncommitted entries within the database.

−z

Copy the database in memory to all database copies. Note that all database copies are in sync with memory and are automatically updated at system shutdown. The -z option lets you update the database copies at your discretion.

EXAMPLES   **EXAMPLE 1**   Creating an AP Database Copy

The following command creates a copy of the AP system database on /dev/rdsk/c2t0d0s1.

  # **apdb −c /dev/rdsk/c2t0d0s1**

SEE ALSO   *Sun Enterprise Server Alternate Pathing 2.3 User Guide*

apconfig(1M), apdisk(1M), apnet(1M)

NAME	apdisk – manage disk path groups
SYNOPSIS	**apdisk** −c −p *primary_disk_controller* −a *physical_disk_controller*
	apdisk −d *primary_disk_controller*
	apdisk −z *primary_disk_controller*
	apdisk −f *physical_disk_controller*
	apdisk −n *physical_disk_controller*
	apdisk −u −p *primary_disk_controller* −a *physical_disk_controller*
	apdisk −w *physical_disk_controller*
DESCRIPTION	The /usr/sbin/apdisk command helps you manage disk path groups in the alternate pathing (AP) system.
OPTIONS	The following options are supported:

−c −p *primary_disk_controller* −a *physical_disk_controller*

Create database entries for disk arrays connected to two I/O ports. Give the I/O port names (for example, sf:0 and sf:1) as the *primary_disk_controller* and *physical_disk_controller*.

Note - Pathnames for disk controllers consist of the I/O port name and the instance number separated by a colon (:). This naming convention differs from previous versions of AP. The addition of the colon (:) delimiter allows AP to provide support for a wider variety of disk controllers. This change does not apply to network pathnames.

−d *primary_disk_controller*

Delete AP information for the specified disk path group. If the existing information is uncommitted, apdisk removes it immediately. If the existing information is already committed, it is only marked for deletion and existing metadevices continue to function until a commit is done, at which time the information is removed.

−z *primary_disk_controller*

Undelete AP information for the specified disk path group. This option cancels a previous apdisk -d request that marked committed information for deletion.

−f *physical_disk_controller*

  Mark the physical disk controller as offline. The corresponding metadisk
  interface can still be used if the other disk controller in the path group is
  functioning properly. Note that you cannot mark a physical disk controller
  as offline if it is currently the active alternate.

−n *physical_disk_controller*

  Mark the physical disk controller as online. Note that this operation does
  not automatically cause the disk controller to become the active alternate.

−u −p *primary_disk_controller* −a *physical_disk_controller*

  Update existing database entries for the disk path group identified by the
  primary path (for example, sf:0). Disk targets that are no longer accessible
  through one or more paths are removed, and new disk targets are added.

  An update can result in the need to update the AP metadevice nodes in
  /dev/ap/[r]dsk. To update the metadevice nodes, use the following two
  commands:

    drvconfig -i ap_dmd

    apconfig -R

  See apconfig(1M) in this manual and drvconfig(1M) in the
  SunOS Reference Manual.

−w *physical_disk_controller*

  Clear the tried flag for the specified I/O controller path.

EXAMPLES   **EXAMPLE 1**   Creating Metadisk Nodes and AP Database Entries

The following commands create metadisk device nodes and AP database
entries for disks that use the pln:0 and pln:1 interfaces, with pln:0
specified as the primary path.

```
# apdisk -c -p pln:0 -a pln:1
# apdb -C
# drvconfig -i ap_dmd
# apconfig -R
```

EXAMPLE 2   Deleting Database Entries

The following commands delete the AP database entries for disks with `sf:1`
specified as the primary path.

```
# apdisk -d sf:1
# apdb -C
```

EXAMPLE 3   Clearing the `Tried` Flag

The following command clears the tried flag for `sf:1`.

```
# apdisk -w sf:1
```

SEE ALSO   *Sun Enterprise Server Alternate Pathing 2.3 User Guide*

apdb(1M), apconfig(1M), apinst(1M), apnet(1M) in this manual and
devlinks(1M), devfsadm(1M)(Solaris 8 only), and drvconfig(1M) in the
SunOS Reference Manual

NAME | apinst – identify disk host adapter instances, /dev/dsk targets

SYNOPSIS | **apinst**

DESCRIPTION | The /usr/sbin/apinst program identifies all disk host bus adapters and provides the *name*:*instance number*, and /dev/dsk targets attached to each.

EXAMPLES
```
% apinst
pln:0
     /dev/dsk/c1t0d0
     /dev/dsk/c1t1d0
     /dev/dsk/c1t2d0
     /dev/dsk/c1t3d0
     /dev/dsk/c1t4d0
     /dev/dsk/c1t5d0
pln:1
     /dev/dsk/c2t0d0
     /dev/dsk/c2t1d0
     /dev/dsk/c2t2d0
     /dev/dsk/c2t3d0
     /dev/dsk/c2t4d0
     /dev/dsk/c2t5d0
sf:0
     /dev/dsk/c3t0d0
     /dev/dsk/c3t1d0
     /dev/dsk/c3t2d0
     /dev/dsk/c3t3d0
     /dev/dsk/c3t4d0
     /dev/dsk/c3t5d0
sf:1
     /dev/dsk/c4t0d0
     /dev/dsk/c4t1d0
     /dev/dsk/c4t2d0
     /dev/dsk/c4t3d0
     /dev/dsk/c4t4d0
     /dev/dsk/c4t5d0
```

NAME | apnet – manage network path groups

SYNOPSIS | **apnet** −c −a *physical_interface* −a *physical_interface*

apnet −d *metanetwork_interface*

apnet −z *metanetwork_interface*

apnet −f *physical_interface*

apnet −n *physical_interface*

apnet −m *metanetwork_interface* −a *physical_interface*

apnet −m *metanetwork_interface* −r *physical_interface*

apnet −t *metanetwork_interface*

apnet −w *metanetwork_interface*

DESCRIPTION | The /usr/sbin/apnet command helps you manage network path groups in the alternate pathing (AP) 2.3 system.

OPTIONS | The following options are supported:

−c −a *physical_interface* **[**−a *physical_interface***]**

Create a metanetwork and network path group for the network connected to the specified network controller paths. If a second −a is given, apnet(1M) designates the specified network interface as the alternate for the metanetwork. (If you initially create a network path group with only one path, you can later add an additional path using apnet− m.)

Note - *metanetwork_interface* refers to an AP metadevice such as mether0. *physical_interface* refers to an actual physical device such as hme0.

Note - For backwards compatibility with older versions of AP the −p option can be used for *physical_interface* instead of −a , however, the −p option has no significance other than indicating an alternate.

−d *metanetwork_interface*

Delete the specified metanetwork and corresponding network path group. If the path group is currently uncommitted, apnet(1M) removes the

metanetwork and the path group immediately. If the path group is committed, the metanetwork and path group are only marked for deletion, and the metanetwork interface continues to function until a commit is performed.

−z *metanetwork_interface*

Undelete the specified metanetwork and path group. This option cancels a previous `apnet -d` request that marked a committed path group for deletion.

−f *physical_interface*

Mark the specified physical interface as offline, making it unavailable to the metanetwork interface.

Note - A physical interface cannot be marked as offline if it is active.

−n *physical_interface*

Mark the specified physical interface as online, making it available to the metanetwork interface.

−m *metanetwork_interface* −a *physical_interface*

Add the physical interface as an alternate path for the specified metanetwork. You can use this option only if there is currently only one interface associated with the metanetwork.

Note - When an alternate is added (−a) or removed (−r) from a committed path group, a commit operation must be performed before the change takes effect. In practice, the existing metanetwork interface is marked for deletion, and a new metanetwork interface is created without affecting usage of the existing interface.

−m *metanetwork_interface* −r *physical_interface*

Remove the physical interface from the specified metanetwork.

> **Note -** When an alternate is added (-a) or removed (-r) from a committed
> path group, a commit operation must be performed before the change takes
> effect. In practice, the existing metanetwork interface is marked for deletion,
> and a new metanetwork interface is created without affecting usage of the
> existing interface.

-t *metanetwork_interface*

Make the next alternate path the temporary active path. This option is
intended for scripts that are trying alternate paths in sequence until a
working path is found. The command returns an error if the sequencing
wraps back to the original primary.

-w *metanetwork_interface*

Make the current temporary active path the actual active path.

EXAMPLES

EXAMPLE 1 Creating a Network Path Group and Metanetwork Interface

The following command creates a network path group and a metanetwork
interface, mether0, which has hme0 and qfe1 as its alternates.

```
# apnet -c -a hme0 -a qfe1
# apdb -C
```

EXAMPLE 2 Deleting a Network Path Group and Metanetwork Interface

The following example deletes the network path group and metanetwork
interface mether0:

```
# apnet -d mether0
# apdb -C
```

SEE ALSO

Sun Enterprise Server Alternate Pathing 2.3 User Guide

apconfig(1M), apdb(1M), apdisk(1M)

Device and Network Interfaces

NAME | Intro – AP special files

DESCRIPTION | This section describes AP files for your Sun Enterprise server.

LIST OF FUNCTIONS

ap**(7D)**   alternate pathing librarian driver, `/dev/ap`

ap_dmd**(7D)**   AP disk metadriver

mether**(7D)**   AP network ethernet metadriver

mfddi**(7D)**   AP network FDDI metadriver

NAME   ap – alternate pathing librarian driver, `/dev/ap`

DESCRIPTION   The AP driver provides a pseudo-driver interface to the kernel alternate pathing (AP) Librarian features.

FILES   The following files are used:

`/kernel/drv/ap`   AP driver module

`/kernel/drv/ap.conf`   AP driver configuration file

SEE ALSO   *Sun Enterprise Alternate Pathing 2.3 User Guide*

`ap_dmd`(7D), `mether`(7D), `mfddi`(7D)

NAME	ap_dmd – AP disk metadriver
SYNOPSIS	**ap_dmd@** *target,lun:partition*
DESCRIPTION	The ap_dmd(7D) driver works with the AP software to support alternate pathing for physical devices handled by the ssd(7D) disk driver. See ssd(7D) in the *SunOS Reference Manual.*

This AP metadriver supports alternate controller paths to a physical device. These paths are associated with a *metadisk device*, which is one of the file system special nodes associated with a particular metadriver.

The ap_dmd(7D) driver enables the AP Librarian, ap(7D), to configure or unconfigure physical controller paths to a disk device by using an interface that allows APSET, APUNSET, and APSWITCH commands. These commands are issued by ap(7D) at the request of the user-invoked AP commands and AP daemon. To change the controller path information associated with a particular ap_dmd(7D) device, use apconfig(1M), apdb(1M) and apdisk(1M). For more information, see the *Sun Enterprise Server Alternate Pathing 2.3 User Guide.*

All device operations supported by the ssd driver are also valid on ap_dmd(7D) devices that have been created by using AP commands. See the other AP commands for details regarding other components of the AP software, and the ssd(7D) man page for information about block/character file accesses, I/O requests, disk partitioning schemes, CD-ROM support, and ioctls.

ERRORS	ENXIO	No physical SCSI path to the target device exists.
	Other	For information on other errors, see ssd(7D).

FILES   The following files are used by this utility:

ap_dmd.conf – driver configuration file

/dev/ap/dsk/m*ncn* t*n* d*n*s*n* – block files

/dev/ap/rdsk/m*ncn* t*n* d*n*s*n* – raw files

where m identifies the device as a metadevice and:

c*n*	Controller number
t*n*	Target number
d*n*	Logical unit number
s*n*	Slice (partition) number

DIAGNOSTICS   See ssd(7D) in the *SunOS Reference Manual.*

SEE ALSO   *Sun Enterprise Server Alternate Pathing User Guide*

apconfig(1M), apdb(1M), ap_disk(1M), apnet(1M), ap_daemon(1M), ap(7D), mether(7D), and mfddi(7D) in this reference manual

NAME	mether – AP network metadriver
SYNOPSIS	`/dev/mether`
DESCRIPTION	The multithreaded, loadable, clonable, STREAMS metanetwork device driver that supports the connectionless Data Link Provider Interface, dlpi(7P), for ethernet drivers. For a list of supported devices see *Release Notes Supplement Solaris 8 2/00.* The mether(7D) driver works with the AP software to support alternate pathing for physical network devices. Device operations of mether(7D) are an extension of the operations of the underlying network drivers. The mether(7D) driver normally operates as a transparent pass-through module; it neither interprets nor modifies any of the STREAMS DLPI type messages. However, it does intercept and modify the DL_ATTACH_REQ and DL_INFO_ACK messages. DL_ATTACH_REQ messages are captured and used to drive the initial connection between logical and physical devices. DL_INFO_ACK messages are captured and responded to with a prebuilt response to eliminate the possibility of the message response timing out due to induced message delays. The cloning, character-special device `/dev/mether` is used to access all device-specific instances of the mether(7D) driver within the system.
mether and AP	The mether driver provides an interface to support Alternate Pathing. The APSET interface enables a user to provide a mapping between physical path and logical path. APUNSET provides an interface to remove a physical-to-logical path mapping, and APSWITCH provides a mechanism to switch a logical path from its existing physical path to a new physical path. For a more complete description of this AP capability, see the *Sun Enterprise Server Alternate Pathing 2.3 User Guide.*
mether and DLPI	The mether(7D) driver is a "style 2" Data Link Service provider. All DLPI processing is handled by the underlying physical device driver. See the man page that corresponds to each underlying driver.
ERRORS	The mether(7D) device driver sets errno as listed for the following conditions:

EBUSY	An attempt was made to unload a busy device or to APUNSET an active device.
EEXIST	An attempt was made to APSET an existing logical-to-physical mapping and a logical path when the system was out of memory.
EIO	An attempt to switch between physical devices failed.

ENODEV   No physical mapping exists.

ENOMEM   System memory was exhausted during an attempt to create a
                  mapping between a physical path and a logical path.

FILES   The following files are used by this utility:

                  `mether.conf` driver configuration file

                  `/dev/mether` metaethernet special character device

DIAGNOSTICS   See `dlpi`(7P) in the *SunOS Reference Manual*.

SEE ALSO   *Sun Enterprise Server Alternate Pathing 2.3 User Guide*

                  `ap_daemon`(1M), `apconfig`(1M), `apdb`(1M), `apnet`(1M), `ap`(7D),
                  `ap_dmd`(7D), and `mfddi`(7D) in this reference manual

                  `driver.conf`(4) and `dlpi`(7P) in the *SunOS Reference Manual*

                  The *SunOS Reference Manual* and other optional reference manuals (for
                  example, the *SunFDDI Reference Manual*), as appropriate.

NAME	mfddi – AP network metadriver for FDDI
SYNOPSIS	`/dev/mfddi`
DESCRIPTION	The multithreaded, loadable, clonable, STREAMS metanetwork device driver that supports the connectionless Data Link Provider Interface, dlpi(7P), for FDDI drivers. For a list of supported devices see *Release Notes Supplement Solaris 8 2/00.*.

The mfddi(7D) driver works with the AP software to support alternate pathing for physical network devices.

Device operations of mfddi(7D) are an extension of the operations of the underlying network drivers. The mfddi(7D) driver normally operates as a transparent pass-through module; it neither interprets nor modifies any of the STREAMS DLPI type messages nor the Sun-specific FDDI messages such as DL_IOC_FDG. However, it does intercept and modify the DL_ATTACH_REQ, and DL_INFO_ACK messages.

DL_ATTACH_REQ messages are captured and used to drive the initial connection between logical and physical devices. DL_INFO_ACK messages are captured and responded to with a prebuilt response to eliminate the possibility of the message response timing out due to induced message delays.

The cloning, character-special device `/dev/mfddi` is used to access all device-specific instances of the mfddi(7D) driver within the system.

mfddi and AP
The mfddi(7D) driver provides an interface to support Alternate Pathing. The APSET interface enables a user to provide a mapping between physical path and logical path. APUNSET provides an interface to remove a physical-to-logical path mapping, and APSWITCH provides a mechanism to switch a logical path from its existing physical path to a new physical path. For a more complete description of this AP capability, see the *Sun Enterprise Server Alternate Pathing 2.3 User Guide*.

mfddi and DLPI
The mfddi driver is a "style 2" Data Link Service provider. All DLPI processing is handled by the underlying physical device driver. See the man page that corresponds to each underlying driver.

ERRORS
The mfddi(7D) device driver sets `errno` as listed for the following conditions:

EBUSY	An attempt was made to unload a busy device or to APUNSET an active device.
EEXIST	An attempt was made to APSET an existing logical-to-physical mapping and a logical path when the system was out of memory.
EIO	An attempt to switch between physical devices failed.

ENODEV   No physical mapping exists.

ENOMEM   System memory was exhausted during an attempt to create a
                    mapping between a physical path and a logical path.

FILES   The following files are used by this utility:

mfddi.conf driver configuration file

/dev/mfddi metaFDDI special character device

DIAGNOSTICS   See dlpi(7P) in the *SunOS Reference Manual*.

SEE ALSO   *Sun Enterprise Server Alternate Pathing 2.3 User Guide*

ap_daemon(1M), apconfig(1M), apdb(1M), apnet(1M), ap(7D),
ap_dmd(7D), and mether(7D) in this reference manual

driver.conf(4) in the *SunOS Reference Manual*

The *SunOS Reference Manual* and other optional reference manuals (for
example, the *SunFDDI Reference Manual*), as appropriate.